PENERAPAN TEORI DASAR DATABASIS RELASIONAL.

Sumartono Prawirosusanto
Laboratorium Elektronika & Instrumentasi,
Jurusan Fisika, FMIPA UGM.

INTISARI.

Databasis relasional, sebagai suatu sistem penyimpanan data dan pengolahan data, disajikan secara singkat. Ciri-ciri dan kelebihan-kelebihan yang dimilikinya dan prasyarat implementasinya juga ditinjau. Kemudian dicoba merencanakan databasis akademis, dimulai dengan menentukan medan-medan data yang diperlukan. Dengan cara penormalan, disusunlah relasi-relasi yang memenuhi bentuk normal ke tiga (3NF). Implementasi dengan bahasa SQL juga ditunjukkan.

PRAKATA.

Teknologi databasis diungkapkan sebagai 'satu bidang yang berkembang paling cepat dalam ilmu komputer dan informasi'. Bidang yang dengan cepat menjadi penting, baik secara praktis maupun secara teoretis. Sekarang dapat dinyatakan bahwa beribu-ribu organisasi menjadi gayut atas operasi terus menerus daripada suatu sistem databasis.

Sistem databasis merupakan sistem penyimpanan rekaman berbasis komputer, dengan maksud utama merekam dan memelihara informasi atau data. Terutama bagi universitas, yang sangat erat hubungannya dengan data-data yang diperlukan untuk pengembangan dan pemantauan sistem-sistem di dalam universitas, misalnya data-data akademis maupun data-data inventarisasi.

Faedah yang dapat diharapkan dari pengembangan penerapan teori dasar Databasis Relasional adalah sangat luas. Databasis relasional bersifat terpadu dan dapat dipakai bersama. Dengan demikian sistem ini akan menyediakan kemudahan dalam pengendalian terpusat data operasional universitas. Keuntungan yang muncul antara lain: pengulangan data dapat dikurangi, ketakserasian data dapat dihindari, data dapat dipakai bersama, pembakuan dapat diterapkan, pembatasan-pembatasan keamanan data dapat diterapkan, keterpaduan dapat dipertahankan dan persyaratan-persyaratan yang bertentangan dapat diseimbangkan.

Untuk universitas data itu akan sangat membantu kelancaran penyelenggaraan pendidikan dan mengurangi pemborosan. Disamping itu kita dapat menambah pengalaman dalam pengelolaan data dengan sistem komputer yang canggih.

1. PENGANTAR.

Sejak E. F. Codd [COD-70] memperkenalkan model databasis relasional, sejumlah sistem prototipe sudah diimplementasikan [Won-79] untuk menunjukkan kemungkinan menopang bahasa-bahasa pengolah-data aras-tinggi dan tak berprosedur yang berlandaskan aljabar relasional ataupun kalkulus relasional. Usaha-usaha selanjutnya telah diarahkan untuk mengimplementasikan sistem yang lebih lengkap.

Suatu sistem databasis relasional yang cukup ideal harus mampu menyediakan fasilitas-fasilitas sebagai berikut [WON-79]:

1) interfas untuk bahasa pengolah data aras-tinggi yang non prosedural yang menyediakan kemampuan untuk pemrograman terapan maupun untuk pemakai bukan teknis dalam : tanya-jawab, pengolah data, dan fasilitas-fasilitas kendali data;
2) struktur file yang efisien untuk penyimpanan databasis dan lintasan capaian yang efisien ke databasis tersimpan;
3) pengoptimisasi yang efisien untuk memenuhi tuntutan waktu-tanggap yang lebih cepat dari pemakai terminal;
4) pandangan (view) dan potret (snapshot) databasis tersimpan oleh pemakai;
5) kendali keterpaduan (kontrol integritas) -validasi kendala-kendala semantik atas databasis selama pengolahan data dan penolakan terhadap pernyataan pengolahan data yang

mengganggu;

6) kendali persaingan (kontrol-konkurensi) -pengaturan saat peremajaan serempak terhadap databasis yang dipakai bersama oleh banyak pemakai;
7) kendali gapaian selektif (kontrol akses selektif) -pemberian hak gapaian terhadap databasis milik satu pemakai kepada pemakai lain;
8) penyembuhan dari kerusakan ringan dan kerusakan berat;
9) pembangkit laporan untuk tampilan yang sangat khas daripada hasil-hasil interaksi terhadap databasis dan fasilitas-fasilitas komputasi yang bersifat terapan, seperti analisis statistik.

Bahasa pengolah data, baik yang berdasarkan aljabar relasional maupun kalkulus relasional, harus lengkap secara relasional dan lengkap secara fungsional. Suatu bahasa dapat dinyatakan lengkap secara relasional jika setiap hubungan yang terdefinisikan dengan ungkapan kalkulus relasional dapat dicari dengan suatu pernyataan di dalam bahasa pengolah data itu [COD-72]. Sedang ia dikatakan lengkap secara fungsional jika ia mendukung permintaan-permintaan pemakai seperti misalnya histogram, pembangkitan laporan, dan fungsi-fungsi gumpalan (maks, min, rerata, jumlah, cacah).

Setelah kita meninjau secara umum ciri-ciri dan kemampuan-kemampuan yang harus dimiliki oleh suatu sistem databasis relasional, pada bab berikut ini kita akan meninjau teori dasar databasis relasional, mulai dari aljabar relasional dan kalkulus relasional, gayutan fungsional, bentuk-bentuk normal, sampai pada penguraian skema relasi, yang kemudian akan diterapkan dalam penyusunan databasis akademis universitas.

2. Teori Pendukung Databasis Relasional.

Dalam teori databasis relasional terdapat tiga pendekatan pokok untuk perancangan bahasa-bahasa yang dipergunakan untuk mengungkapkan pertanyaan-pertanyaan mengenai relasi-relasi (hubungan-hubungan atau tabel-tabel). Notasi untuk pengungkapan pertanyaan-pertanyaan merupakan bagian paling nyata dari bahasa pengolah data (DML, data manipulation language). Sisi bukan pertanyaan bahasa pengolah data relasional atau "bahasa pertanyaan", yalah peremajaan data yang berkaitan dengan penyisipan, penghapusan dan perubahan tupel-tupel.

Bahasa-bahasa pertanyaan model relasional terpecah ke dalam dua kelas besar:

1. Bahasa aljabar, dengan pertanyaan-pertanyaan diungkapkan dengan penerapan operator-operator khusus terhadap relasi,
2. Bahasa kalkulus predikat, dengan pertanyaan-pertanyaan melukiskan himpunan tupel-tupel dengan menetapkan suatu predikat yang harus dipenuhi tupel-tupel tersebut.

Yang kedua ini dibagi lagi ke dalam dua kelas, tergantung pada obyek-obyek primitifnya adalah tupel atau unsur ranah ('domain') atribut tertentu [DAT-81; DEL-85; ULL-82].

2.1. Aljabar Relasional.

Operator dan Operand dalam Aljabar Relasional.

Suatu relasi adalah himpunan n-tupel untuk tetapan n tertentu, yang disebut derajat relasi tersebut. Sering lebih memudahkan bila kita memberi nama pada komponen tupel, yang merupakan atribut relasi itu. Apabila tidak diberi nama maka urutan kolom menjadi penting. Dalam penanganan relasi-relasi sebagai suatu databasis diandaikan bahwa mereka berhingga. Kendala keberhinggaan ini memang menimbulkan sedikit kesukaran dalam definisi aljabar relasional maupun kalkulus relasional.

Sekedar contoh : kita tidak dapat mempunyai operasi aljabar komplemen, karena NOT R menyatakan relasi tak berhingga : himpunan semua tupel yang tidak ada di dalam R. Tidak ada cara untuk mendaftar tupel-tupel isi relasi NOT R.

Operand aljabar relasional adalah relasi-relasi tetap atau peubah yang menyatakan relasi dengan derajat tertentu. Ada lima operasi dasar yang dapat dipakai mendefinisikan aljabar relasional:

1. Union.

Union relasi R dan S, dinyatakan dengan R+S, adalah himpnan tupel yang ada di dalam R atau di dalam S atau di dalam ke dua-

duanya. Operasi union dapat diterapkan hanya bila ke dua relasi mempunyai derajat yang sama, sehingga hasilnya juga mempunyai derajat (cacah komponen) yang tetap sama.

2. **Selisih himpuan.**

Selisih antara R dan S, dinyatakan dengan R-S, adalah himpunan tupel-tupel yang berada di dalam R tetapi tidak ada di dalam S. Di sini ke dua relasi juga harus mempunyai derajat yang sama.

3. **Darab Cartes (Produk Cartesan).**

Bila R dan S adalah relasi dengan derajat k1 dan k2, maka RxS, darab Cartes dari R dan S, adalah himpunan (k1 + k2)-tupel dengan k1 komponen adalah tupel dari R dan k2 komponen belakang adalah tupel dari S.

4. **Proyeksi.**

Operasi ini mengambil beberapa atau sebagian komponen tupel R dan/atau menyusunnya menjadi komponen relasi yang lebih kecil. Jika R(A1, A2, ..., Ak) adalah relasi dengan derajat k, maka R[X] dengan atribut X merupakan subhimpunan daripada himpunan atribut {A1, A2, ..., Ak} dan Y adalah himpunan atribut sisanya, menyatakan proyeksi R ke dalam komponen-komponen X, yaitu himpunan tupel-x sedemikian sehingga terdapat k-tupel xy di dalam R.

5. **Seleksi.**

Misalkan F adalah suatu rumus yang mengandung :

i) operand yang berupa tetapan atau nomor komponen atau nama atribut,

ii) operator perbandingan <, =, >, <=, >= dan =/=.

iii) operator nalaran AND, OR, dan NOT.

R[F] adalah himpunan tupel t di dalam R yang memenuhi persyaratan bahwa bila kita sulihkan pada komponen x dari tupel t untuk semua pemunculan x dalam rumus F, rumus F menjadi benar.

Misalnya R[A2>A3] menyatakan himpunan tupel di dalam R dengan komponen ke dua lebih besar daripada komponen ke tiga.

Beberapa Operasi Aljabar tambahan.

Ada beberapa operasi yang berguna yang dapat diungkapkan dengan operasi-operasi di atas, tetapi karena manfaatnya mereka telah diberi nama sendiri dan sering juga digunakan sebagai operasi sederhana (primitif).

1. **Saling-Sayat (Interseksi).**
   R INT S adalah singkatan untuk R - (R - S).
2. **Kuosien (Hasilbagi).**
   Bila R dan S adalah relasi dengan derajat r dan s, dengan r>s dan S=/=0. Maka R/S adalah himpunan (r-s)-tupel t sedemikian sehingga untuk semua s-tupel u di dalam S, tupel t u ada di dalam R.
3. **Join.**
   Join-O dari R dan S pada atribut ke-i dari R dan atribut ke-j dari S, ditulis R[Ai O Aj]S, dengan O operator perbandingan (=, <, dan seterusnya), adalah singkatan dari (RxS)[Ai O A(r+j)], jika R berderajat r. Jadi : join-O dari R dan S adalah tupel dalam darab Cartes dari R dan S yang sedemikian sehingga komponen ke-i dari R berhubungan O dengan komponen ke-j dari S.
   Jika O adalah =, operasi ini disebut ekuijoin.
4. **Join alami (join natural).**
   Ditulis R*S, berlaku bila R dan S atributnya diberi nama.
   Mencarinya adalah sebagai berikut:
   i) Hitung RxS.
   ii) Untuk setiap atribut A yang ada di dalam R dan di dalam S, maka diambil tupel-tupel dari RxS yang mempunyai R.A dan S.A sama.
   iii) Untuk setiap atribut A di atas, dihapus salah satu (misalnya S.A).

2.2. Kalkulus Relasional.

Ungkapan dalam kalkulus relasional tupel adalah dalam bentuk {t| F(t)}, dengan t peubah tupel, yaitu peubah yang menyatakan tupel dengan panjang tertentu dan F adalah rumus yang disusun dari atom-atom dan kumpulan operator, yang didefinisikan dibawah ini.
Atom-atom rumus F ada tiga tipe, sebagai berikut :
1. R(s), dengan R nama relasi dan s peubah tupel. Atom ini menyatakan atau menegaskan bahwa s adalah tupel di dalam relasi R.
2. s[i] O u[j], dengan s dan u peubah tupel dan O adalah operator perbandingan (<, =, dan seterusnya). Atom ini menyatakan bahwa komponen ke-i daripada s berbandingan O dengan komponen ke-j

daripada u. Misal, s[1] < u[3] berarti komponen pertama s kurang daripada komponen ketiga u.

3. s[i] O a dan a O s[i] dengan O seperti pada nomor 2 di atas, dan a adalah tetapan. Ungkapan yang pertama menyatakan bahwa komponen ke-i daripada s berbandingan O dengan tetapan a. Ungkapan yang kedua mempunyai arti yang serupa.

Pengertian peubah tupel 'bebas' dan 'terikat', dalam definisi operator-operator kalkulus relasional, dapat dianalogkan seperti di dalam bahasa pemrograman. 'Peubah bebas' adalah analog dengan peubah global dalam bahasa pemrograman, dan 'peubah terikat' seperti peubah lokal (setempat), yang didefinisikan di dalam prosedur atau subrutin yang ditinjau dan tidak dapat diacu dari luar,yaitu dari program utama atau prosedur yang lain.

Rumus-rumus, pemunculan bebas dan terikat peubah-peubah tupel dalam rumus-rumus itu, didefinisikan secara rekursif sebagai berikut :

1. Setiap atom adalah rumus.

   Semua pemunculan peubah tupel yang disebut dalam atom adalah bebas di dalam rumus ini.

2. Jika F1 dan F2 adalah rumus, maka F1 AND F2, F1 OR F2 dan NOT F1 adalah rumus-rumus yang menegaskan 'F1 dan F2 keduanya benar', 'F1 atau F2, atau kedua-duanya, benar', dan 'F1 tidak benar'. Peubah tupel adalah bebas atau terikat, seperti apa yang terdapat di dalam F1 atau F2, bebas atau terikat.

3. Jika F adalah rumus, maka **(exist** s)(F) adalah rumus.

   Pemunculan s yang bebas di dalam F menjadi terikat pada (**exist** s) di dalam (**exist** s)(F). Rumus ini menegaskan bahwa ada suatu harga s sedemikian sehingga bila kita sisipkan atau sulihkan harga ini pada semua pemunculan bebas s di dalam F, maka rumus F menjadi benar.

4. Jika F adalah rumus, maka (**all** s)(F) adalah rumus.

   Pemunculan bebas s di dalam F menjadi terikat pada (**all** s) di dalam (**all** s)(F). Rumus ini menegaskan bahwa sebarang harga s dengan derajat yang benar disulihkan pada pemunculan bebas s di dalam F, maka rumus F itu menjadi benar.

5. Tanda kurung dapat ditambahkan pada rumus seperlunya menurut kebutuhan.

Kita andaikan bahwa urutan penting operator adalah : tertinggi operator perbandingan, lalu penegas **exist** dan **all**, lalu **NOT**, **AND** dan **OR**.

6. Tidak ada bentuk rumus yang lain.

Contoh:

Union daripada R dan S diungkapkan dengan ungkapan kalkulus

{t | R(t) **OR** S(t)}

Atau dinyatakan dengan kata-kata sebagai berikut: union adalah 'himpunan tupel t sedemikian sehingga t ada di dalam R atau di dalam S'.

Selisih R-S diungkapkan oleh {t | R(t) **AND NOT** S(t)}.

Darab RxS diungkapkan oleh :

{t(r+s)| (**exist** u(r))(**exist** v(s))(R(u) **AND** S(v))
AND t[1]=u[1] **AND** ... **AND** t[r]=u[r]
AND t[r+1]=v[1] **AND** ... **AND** t[r+s]=v[s])}

Disini t(i) menyatakan tupel t berderajat i.

Proyeksi R[i1, i2, ..., ik] diungkapkan oleh

{t(k)| (**exist** u)(R(u) **AND** t[1]=u[i1] **AND** ... **AND**
t[k]=u[ik])}

Seleksi R[F] dinyatakan dengan {t| R(t) **AND** F'}, disini F' merupakan rumus F dengan setiap operand i yang menyatakan komponen ke-i, diganti dengan t[i].

Terlihat disini bahwa ungkapan-ungkapan aljabar relasional dapat dinyatakan dalam ungkapan kalkulus relasional tupel.

2.3. Gayutan berfungsi (Dependensi Fungsional).

Relasi dapat dipakai untuk memodel 'dunia nyata' dalam berbagai cara : misalnya, setiap tupel relasi dapat menyajikan suatu ujud (entitas) dengan atribut-atributnya, atau menyajikan hubungan di antara ujud-ujud (entitas) tersebut. Dari kenyataan bahwa tidak semua tupel dapat menjadi anggota relasi pada satu saat, kita dapat membedakan dua macam pembatasan pada relasi.

1. Pembatasan yang gayut pada semantika unsur-unsur ranah (domain).

Pembatasan ini gayut atas pengertian kita tentang arti tupel. Misalnya, tidak ada seseorang yang mempunyai tinggi 150 meter. Ini akan merupakan kendala keterpaduan.

2. Pembatasan pada relasi yang hanya gayut pada kesamaan atau ketaksamaan nilai-nilai.

Misalkan R(A1, A2, ..., An) suatu skema relasi, dan ambillah X dan Y subhimpunan daripada {A1, A2, ..., An}. Kita katakan X --> Y, yang dibaca `X secara berfungsi menentukan Y' atau `Y secara berfungsi gayut pada X', jika relasi r adalah harga saat ini daripada R, maka bila terdapat dua tupel dari r yang cocok dalam komponen dengan atribut di dalam himpunan X maka mereka pasti cocok dalam sebagian komponen dengan atribut di dalam himpunan Y.

Satu-satunya cara untuk menentukan gayutan berfungsi (dependensi fungsional) yang berlaku untuk skema relasi R adalah dengan meninjau secara cermat apa arti atribut-atributnya. Dalam hal demikian, gayutan (ikatan) merupakan penegasan tentang dunia nyatanya; ia tak dapat dibuktikan tetapi kita dapat mengharapkan bahwa hal itu dilaksanakan oleh DBMS apabila telah dinyatakan oleh perancang databasis. Kebanyakan sistem yang ada akan melaksanakan gayutan-gayutan berfungsi itu sebagai akibat dari fakta bahwa kunci menentukan atribut-atribut lain relasi tersebut. Dengan ditegaskannya bahwa gayutan berfungsi berlaku maka implementasi relasi yang lebih efisien menjadi dimungkinkan.

Contoh : Skema relasi

DAFMAKU (KODEMK, NAMAMK, NIPDOS, NAMADOS, SKS, SEM, PROSTUDI)

Di dalamnya berlaku gayutan berfungsi (Functional Dependency, FD) dari Gambar 3-2:

NAMADOS ---> KODEMK

```
KODEMK   ---> NAMAMK, SKS, SEM
NIPDOS   ---> NAMADOS
PROSTUDI ---> KODEMK
```

Gayutan di atas menyatakan bahwa seorang staf pengajar menentukan matakuliah-matakuliah yang diajarkannya, kode matakuliah menentukan nama matakuliah, jumlah SKS dan pada semester ke berapa matakuliah tersebut ditawarkan atau diberikan. NIP seorang dosen menentukan namanya, dan nama program studi menentukan nomor kode matakuliah yang termasuk di dalam program studi itu.
Di samping itu ada gayutan fungsional trivial yang berlaku

```
NAMADOS        ---> NAMADOS
KODEMK, NAMAMK ---> NAMAMK
```

Juga ada gayutan fungsional taktrivial yang menyertai gayutan-gayutan fungsional di atas misalnya :

```
KODEMK, NIPDOS ---> NAMAMK, NAMADOS
```

Gayutan ini menyatakan bahwa gabungan kode matakuliah dan NIP dosen menentukan nama matakuliah dan nama dosen. Hal ini disebabkan oleh karena NAMAMK ditentukan oleh KODEMK dan NAMADOS ditentukan oleh NIPDOS.

Implikasi nalar Gayutan berFungsi (Dependensi Fungsional).
Andaikan R adalah suatu skema relasi dan A, B, dan C adalah beberapa atributnya. Andaikan juga bahwa gayutan-gayutan fungsional A --> B dan B --> C berlaku di dalam R. Kita nyatakan bahwa A --> C harus juga berlaku di dalam R.
Secara umum, misalkan F adalah suatu himpunan gayutan fungsional (FD) untuk skema relasi R, dan misalkan X --> Y adalah gayutan berfungsi. Kita katakan F secara nalar mengimplikasikan X --> Y, dituliskan F |= X --> Y, jika setiap relasi r dari R yang memenuhi gayutan-gayutan berfungsi di dalam F juga memenuhi X --> Y. Jika F mengandung A --> B dan B --> C, maka A --> C secara nalar diimplikasikan oleh F. Yaitu, {A --> B, B --> C} |== A --> C.

F+ , lingkupan (closure) daripada F, adalah himpunan gayutan-gayutan berfungsi yang secara nalar diimplikasikan oleh F, yaitu

F+ = { X --> Y | F |== X --> Y }

Contoh :

Misalkan R = ABC dan F = {A --> B, B --> C}.
Maka himpunan F terdiri dari semua gayutan X --> Y yang memenuhi salah satu dari yang berikut:

1. X mengandung A, seperti : ABC --> AB, AB --> BC, atau A --> C.
2. X mengandung B tetapi tidak mengandung A, dan Y tidak mengandung A, seperti : BC --> B, B --> C, atau B --> 0.
3. X --> Y adalah satu dari dua gayutan C --> C atau C --> 0.

Kunci.

Jika R adalah suatu skema relasi dengan atribut-atribut A1, A2, bagian daripada {A1, A2, ..., An}, kita katakan X adalah kunci daripada R jika : X --> A1, A2, ..., An ada di dalam F+. Yaitu gayutan yang menyatakan semua atribut gayut pada atribut X diberikan atau terturunkan secara nalar dari gayutan yang diberikan. Sedangkan untuk semua subhimpunan Y, yang merupakan subhimpunan sejati daripada himpunan X, Y --> A1, A2, ..., An tidak termasuk di dalam F+.

Aksiom untuk Gayutan berFungsi.

Untuk menghitung F+ dari F, atau, diberikan F dan gayutan berfungsi X --> Y, bagaimana menentukan bahwa X --> Y ada di dalam F+, diperlukan kaidah-kaidah inferensi. Kaidah ini dinamakan aksiom-aksiom Armstrong [ARM-74]. Kita andaikan diberikan skema relasi dengan himpunan atribut U, himpunan semesta (universal) atribut-atribut, dan himpunan gayutan berfungsi F yang hanya menyangkut atribut-atribut di dalam U.

Kaidah-kaidah tersebut adalah :

A1. (refleksivitas)

Jika Y himpunan-bagian dari X himpunan-bagian dari U, maka X --> Y secara nalar diimplikasikan oleh F. Kaidah ini memberikan gayutan-gayutan trivial, yaitu gayutan yang ruas kanannya terkandung pada ruas kiri, dan penggunaannya hanya gayut pada U, dan tidak pada F.

A2. (perluasan, augmentasi)

Jika X --> Y berlaku, dan Z himpunan-bagian dari U, maka XZ --> YZ berlaku. Disini, X, Y dan Z adalah himpunan atribut, dan XZ berarti X+Z (Union). Gayutan yang diberikan X --> Y, mungkin sudah ada di dalam F atau dapat diturunkan dari gayutan-gayutan di dalam F.

A3. (transitivitas)

Jika X --> Y dan Y --> Z berlaku, maka X --> Z berlaku.

Contoh :

```
NAMADOS --> KODEMK               (diberikan)
KODEMK  --> NAMAMK, SKS, SEM     (diberikan)
NAMADOS --> NAMAMK, SKS, SEM     (transitivitas).
```

Kaidah lain yang dapat diturunkan dari dan/atau dibuktikan dengan kaidah-kaidah di atas adalah kaidah inferens berikut:

K1. Kaidah penggabungan (union).

{ X --> Y, X --> Z } |== X --> YZ.

K2. Kaidah transitivitas semu (pseudo-transitivitas).

{ X --> Y, WY --> Z } |== XW --> Z.

K3. Kaidah penguraian (dekomposisi).

Jika X --> Y berlaku, dan Z himpunan-bagian dari Y, maka X --> Z berlaku.

Bukti bahwa aksiom Armstrong adalah benar dan lengkap disajikan oleh Ullman [ULL-82], demikian juga algoritme untuk menghitung lingkupan (closure) F+.

2.4. Bentuk-bentuk Normal untuk Skema Relasi.

Bentuk Normal pertama (1NF) dan Bentuk Normal kedua (2NF).

Suatu skema relasi dikatakan ada dalam bentuk normal pertama

(1NF) jika dan hanya jika semua ranah penyangga hanya mengandung nilai-nilai atomik, dan dengan demikian nilai-nilai atribut juga atomik dan bukan merupakan gabungan beberapa nilai.

Suatu skema relasi dikatakan ada dalam bentuk normal kedua (2NF) jika dan hanya jika ia ada dalam bentuk 1NF dan setiap atribut bukan kunci gayut penuh padaa kunci utama.

Sejumlah sifat, atau `bentuk normal' untuk skema relasi dengan gayutan-gayutan fungsional telah didefinisikan. Yang paling signifikan dinamakan `bentuk normal ketiga' (3NF) dan `bentuk normal Boyce Codd' (BCNF). Bentuk-bentuk normal ini menjamin bahwa sebagian besar masalah pengulangan data dan kelainan (anomali) yang dibicarakan dalam pasal 1 tidak akan muncul.

Bentuk Normal Boyce-Codd.

Bentuk-bentuk normal itu yang paling kuat dinamakan Bentuk Normal Boyce Codd (BCNF). Suatu skema relasi R dengan gayutan-gayutan F dikatakan ada dalam bentuk normal Boyce Codd jika X --> A berlaku di dalam R, dan A tidak di dalam X. X adalah adikunci (superkunci) untuk R, yaitu, X merupakan kunci atau mengandung kunci. Dengan kata lain, satu-satunya gayutan tak trivial adalah gayutan-gayutan yang di dalamnya kunci menentukan secara berfungsi satu atau lebih atribut-atribut yang lain.

Contoh:

Skema relasi KAP (dengan K kota, A alamat dan P kodepos), dengan gayutan KA --> P dan P --> K, tidak berada dalam bentuk normal Boyce Codd (meskipun memenuhi bentuk normal ketiga). Alasannya adalah bahwa P --> K berlaku (malah merupakan gayutan yang diberikan), tetapi P bukan kunci daripada KAP, dan juga ia tidak mengandungi kunci.

Bentuk Normal Ketiga (3NF).

Ternyata, dalam beberapa keadaan, bentuk normal Boyce Codd merupakan suatu syarat gayutan yang terlalu keras, dalam arti bahwa ia tidak memungkinkan kita menguraikan suatu skema relasi ke dalam bentuk itu tanpa kehilangan sifat mempertahankan gayutan-gayutan fungsional. Karenanya bentuk normal ketiga (3NF)

dapat dilihat sebagai suatu syarat yang mempunyai keuntungan BCNF dalam menghilangkan kelainan, dan disamping itu kita dapat mengubah ke skema databasis lain tanpa kehilangan sifat mempertahankan gayutan berfungsi. Ia juga memiliki sifat join nirlesap.
Untuk mendefinisikan bentuk normal ketiga diperlukan pengertian atribut prima. Kita katakan atribut A dalam skema relasi R sebagai atribut prima jika A adalah anggota dari salah satu kunci untuk R (bila R mempunyai banyak kunci). Jika A tidak demikian, maka A bukan prima.

Contoh.
Dalam skema relasi KAP, semua atribut adalah prima, karena gayutan-gayutan yang diberikan KA --> P dan P --> K, maka KA dan AP kedua-duanya adalah kunci.
Dalam skema relasi ABCD dengan gayutan AB --> C, B --> D, dan BC --> A kita dapat mengecek bahwa AB dan BC adalah kunci, sehingga A, B dan C adalah prima dan D bukan prima.
Suatu skema relasi R ada dalam **bentuk normal** ketiga jika X --> A berlaku dalam R , maka X adalah adikunci (superkunci) untuk R bila A tidak termasuk di dalam X, atau A adalah prima.
Jika X --> A menyalahi bentuk normal ketiga, maka salah satu dari kasus berikut harus muncul :
1. X adalah subhimpunan sejati dari suatu kunci, atau
2. X adalah subhimpunan sejati dari bukan kunci.
Dalam kasus pertama, dikatakan bahwa X --> A adalah gayutan parsial (perangan), dan dalam kasus kedua ia dinamakan gayutan transitif. Jika R tidak mempunyai gayutan-gayutan perangan (parsial), meskipun ia mungkin mempunyai gayutan-gayutan transitif, dikatakan bahwa R ada dalam bentuk normal kedua (2NF).

2.5. Penguraian skema Relasi.

Penguraian skema relasi R(A1, A2 ,..., An) adalah penggantian dengan kumpulan p = {R1, R2,..., Rk} daripada subhimpunan-subhimpunan R sedemikian sehingga

R = R1 + R2 + ... + Rk.

Tak ada persyaratan bahwa Ri harus saling-asing (disjoin). Salah satu alasan untuk melakukan penguraian yalah bahwa penguraian ini dapat menghilangkan beberapa masalah yang timbul dalam pasal 1.

Join Nirlesap (Lossless Join).

Jika R suatu skema relasi teruraikan ke dalam skema relasi R1, R2, ..., Rk , dan D suatu himpunan gayutan fungsional (FD), maka dikatakan bahwa penguraian itu merupakan penguraian join nirlesap (lossless join decomposition) jika untuk setiap relasi r dari R yang memenuhi D berlaku :

r = r[R1] * r[R2] * ... * r[Rk],

yaitu r merupakan join alami (join natural) dari proyeksi-proyeksinya ke dalam semua Ri.

Jika p = { R1, R2, ..., Rk }, maka pemetaan mp(r) yang didefinisikan sebagai mp(r) = * r[Ri] (i berjalan dari 1 sampai k), yaitu mp(r) adalah join daripada proyeksi-proyeksi r ke dalam skema relasi di dalam p. Dengan demikian, syarat join nirlesap terhadap himpunan gayutan berfungsi D dapat diungkapkan sebagai berikut : untuk semua r yang memenuhi D, r = mp(r).

Lemma 1:

Misalkan R adalah suatu skema relasi, p = (R1, R2,..., Rk) adalah suatu penguraian R, r relasi untuk R, dan ri = r[Ri], maka :

a) r himpunan-bagian dari mp(r).

b) jika s = mp(r), maka s[Ri] = ri.

c) mp(mp(r)) = mp(r).

Algoritme untuk mengetes join nirlesap dapat ditemui dalam [ULL-82], lengkap dengan buktinya.

Teorem I.

Jika p = (R1, R2) adalah suatu penguraian R, dan F adalah himpunan gayutan berfungsi, maka p mempunyai join nirlesap terhadap F jika dan hanya jika (R1 INT R2) --> (R1 - R2) atau (R1 INT R2) --> (R2 - R1).

Perlu dicatat bahwa gayutan-gayutan berfungsi ini tidak harus ada di dalam F, tetapi cukup bila gayutan itu termasuk di dalam F+.

Contoh:

Andaikan R = ABC dan F = { A --> B }. Maka penguraian R ke dalam AB dan AC mempunyai join nirlesap, karena AB INT AC = A, AB - AC = B, dan A --> B berlaku. Tetapi jika kita menguraikan R ke dalam R1 = AB dan R2 = BC, ditemukan bahwa R1 INT R2 = B, dan B secara fungsional tidak menentukan R1 - R2 = A maupun R2 - R1 = C. Sehingga penguraian AB dan BC tidak mempunyai join nirlesap terhadap F = { A --> B}, yang juga dapat dilihat dengan meninjau relasi r = {a1 b1 c1, a2 b1 c2} untuk R. Maka r[AB] = {a1 b1, a2 b1}, r[BC] = {b1 c1, b1 c2} dan r[AB] * r[BC] = {a1 b1 c1 , a1 b1 c2, a2 b1 c1, a2 b1 c2}, tidak sama dengan r.

Penguraian yang mempertahankan Gayutan-gayutan berFungsi.

Kita telah melihat bahwa untuk suatu skema relasi diinginkan suatu penguraian yang mempunyai sifat join nirlesap, karena ia menjamin bahwa setiap relasi dapat ditemukan kembali dari proyeksi-proyeksinya. Sifat penting yang lain daripada suatu penguraian skema relasi R ke dalam p = {R1, ..., Rk} yalah bahwa himpunan gayutan berfungsi F dari R terimplikasikan oleh proyeksi-proyeksi F dalam semua Ri. Secara formal, proyeksi F ke dalam suatu himpunan atribut Z, dinyatakan dengan F[Z], adalah himpunan gayutan berfungsi X --> Y di dalam F sedemikian sehingga XY himpunan-bagian dari Z. Perlu dicatat bahwa X --> Y tidak perlu ada di dalam F, tetapi harus termasuk di dalam F+.

Kita katakan penguraian p tidak mengubah (atau dengan kata lain mempertahankan) himpunan gayutan F jika union (gabungan) semua gayutan berfungsi di dalam F[Ri], untuk i = 1, 2,...,k , secara nalar mengimplikasikan semua gayutan berfungsi di dalam F.

Alasan diinginkannya p mempertahakan F yalah bahwa gayutan berfungsi dalam F dapat dipandang sebagai kendala keterpaduan untuk relasi R. Jika gayutan-gayutan terproyeksi tidak mengimplikasikan F, maka bila kita menyajikan R dengan p = {R1, ..., Rk}, kita dapat menemui harga sesaat semua Ri, sebagai penyaji relasi R, yang tidak memenuhi F, meskipun p mempunyai sifat join nirlesap terhadap F. Dengan kata lain, setiap peremajaan salah satu Ri akan memerlukan suatu join untuk mengecek agar kendala-kendala itu tidak dilanggar.

Perlu dicatat bahwa suatu penguraian mungkin mempunyai join nirlesap terhadap himpunan gayutan berfungsi F, tetapi tidak mempertahankan F. Sebaliknya, penguraian skema relasi dapat mempertahankan F tetapi tidak mempunyai join nirlesap. Contoh dapat dilihat dalam [ULL-82].

Penguraian Join nirlesap ke dalam Bentuk Normal Boyce-Codd (BCNF)

Kita telah melihat dua sifat utama skema databasis sebagai keseluruhan, yalah sifat join nirlesap dan sifat mempertahankan gayutan-gayutan berfungsinya. Ternyata bahwa sebarang skema relasi mempunyai penguraian ke dalam bentuk normal Boyce-Codd (BCNF) dengan join nirlesap, dan ia mempunyai penguraian ke dalam bentuk normal ketiga yang memiliki join nirlesap dan juga mempertahankan gayutan berfungsinya. Tetapi ada kemungkianan bahwa penguraian suatu skema relasi ke dalam bentuk normal Boyce-Codd yang sekaligus mempertahankan gayutan-gayutan berfungsinya tidak didapatkan. Contoh khas adalah skema relasi KAP. Ia tidak berada di dalam BCNF karena berlakunya gayutan berfungsi P ---> K, meskipun demikian jika kita menguraikan KAP ke dalam sebarang uraian asalkan KAP bukan merupakan salah satu skemanya dalam penguraian, maka gayutan berfungsi KA ---> P tidak dapat diimplikasikan oleh gayutan-gayutan berfungsi terproyeksi.

Lemma 2:

a. Andaikan R suatu skema relasi dengan gayutan-gayutan fungsional F. Misalkan p = {R1, ..., Rk} adalah penguraian R dengan join nirlesap terhadap F. Untuk suatu i khusus, misalkan Fi = F[Ri], dan misalkan q = {S1, ..., Sm} adalah penguraian Ri yang joinnya nirlesap terhadap Fi. Maka penguraian R ke dalam {R1, ..., R(i-1), S1, ..., Sm , R(i+1), ..., Rk} mempunyai join nirlesap terhadap F.

b. Andaikan R, F dan p seperti dalam (a) di atas, dan misalkan r = { R1, ..., Rk, R(k+1), ..., Rn} adalah penguraian R ke dalam himpunan skema relasi yang mengandungi relasi yang ada di p. Maka r juga mempunyai join nirlesap terhadap F.

Algoritme: Penguraian (dengan) Join Nirlesap ke dalam Bentuk Normal Boyce-Codd (BCNF).

Masukan: Skema relasi R dan gayutan fungsional F.

Keluaran: Penguraian daripada R dengan join nirlesap, sedemikian sehingga setiap skema relasi dalam penguraian itu ada dalam bentuk normal Boyce-Codd (BCNF) terhadap proyeksi F ke dalam skema itu.

Metode: Kita secara iteratif menyusun penguraian p untuk R. Dalam setiap saat, p akan memiliki join nirlesap terhadap F. Mula-mula, p terdiri dari R sendiri. Jika S adalah suatu skema relasi dalam p, dan S tidak dalam BCNF, misalkan X --> A adalah gayutan yang berlaku di dalam S, dengan X bukan adikunci (super-kunci) untuk S, dan A tidak di dalam X. Gantikanlah S di dalam p dengan S1 dan S2, dengan S1 terdiri dari A dan atribut-atribut X, dan S2 terdiri dari semua atribut S kecuali A. S2 dengan demikian merupakan suatu subhimpunan sejati daripada S. S1 juga merupakan subhimpunan sejati, atau bila bukan maka X = S - A, sehingga X adalah adikunci untuk S.

Dengan teorem di atas, penguraian S ke dalam S1 dan S2 mempunyai join nirlesap terhadap F himpunan gayutan terproyeksi ke dalam S, karena S1 INT S2 = X , S1 - S2 = A, dan dengan demikian (S1 INT S2) --> (S1 - S2). Dengan lemma-1 (a) di atas, p (dengan S digantikan S1 dan S2) memiliki join nirlesap, jika p juga memiliki join nirlesap. Karena S1 dan S2 masing-masing mempunyai atribut yang lebih sedikit daripada S, dan setiap skema relasi dengan dua atribut atau kurang tentu berada dalam bentuk normal Boyce-Codd (BCNF), kita akhirnya mencapai titik atau keadaan dengan skema relasi di dalam p ada dalam bentuk normal Boyce-Codd (BCNF). Pada saat itu, p masih tetap memiliki join nirlesap, karena p awal yang terdiri dari R sendiri memilikinya dan dalam setiap perubahan p mempertahankan sifat join nirlesapnya.

Contoh:

Marilah kita meninjau skema relasi KDJRMN, dengan K = matakuliah, D = Dosen, J = jam, R = ruang, M = mahasiswa, dan N = nilai ujian. Gayutan-gayutan fungsional F yang diandaikan adalah

K --> D setiap matakuliah hanya mempunyai satu dosen, JR --> K hanya satu matakuliah dapat berlangsung dalam satu ruang pada satu waktu,

JD --> R seorang dosen hanya dapat berada di dalam satu ruang pada satu waktu,

KM --> N setiap mahasiswa mempunyai satu nilai dalam setiap matakuliah,

JM --> R seorang mahasiswa hanya dapat berada di satu ruang pada satu waktu.

Satu-satunya kunci untuk KDJRMN yalah JM. Untuk menguraikan skema relasi ini ke dalam bentuk normal Boyce-Codd, dilakukan beberapa tahapan dan akhirnya di dapat penguraian akhir dari KDJRMN yalah KMN, KD, KJR dan KJM. Ke empat skema relasi ini mentabulasikan

a) nilai matakuliah untuk mahasiswa

b) dosen dari setiap matakuliah

c) jam masing-masing matakuliah diberikan dan ruang untuk setiap jamnya, dan

d) jadwal matakuliah dan jam untuk setiap mahasiswa.

Penguraian di atas di dapat sebagai berikut :

Ditinjau gayutan KM --> N, yang bertentangan dengan syarat BCNF, karena KM tidak mengandung kunci. Jadinya, dengan algoritme penguraian di atas, kita pertama-tama menguraikan KDJRMN ke dalam KMN dan KDJRM. Untuk penguraian selanjutnya kita harus menghitung F+ dan memproyeksikannya ke dalam KMN dan KDJRM. Proses ini umumnya memakan waktu, karena besar F+ dapat eksponensial dalam besar F. Ia mempunyai semua gayutan fungsional trivial yang didapat dengan refleksivitas dan, sebagai tambahan daripada yang ada di dalam F, beberapa gayutan fungsional taktrivial seperti KJ --> R, JM --> K, dan JR --> D. Sekali kita mendapatkan F+, kita pilih mereka yang hanya menyangkut K, M, dan N. Ini adalah F[KMN]. Himpunan ini mempunyai penutup minimal yang terdiri daripada KM --> N sendiri; gayutan-gayutan fungsional yang lain dalam himpunan mengikuti gayutan fungsional ini dari aksiom Armstrong. Kita juga memproyeksikan F+ ke dalam KDJRM.

F[KDJRM] mempunyai penutup minimal

K --> D DJ --> R

JR --> K JM --> R

dan satu-satunya kunci untuk KDJRM adalah JM.

Adalah mudah untuk mencek bahwa KMN ada dalam bentuk normal Boyce-Codd terhadap gayutan-gayutan fungsional terproyeksinya. KDJRM harus diuraikan lebih lanjut, dan kita dapat memilih gayutan fungsional K --> D untuk memecahnya ke dalam KD dan KJRM. Penutup minimal untuk gayutan-gayutan fungsional terproyeksi adalah K --> D untuk KD dan KJ --> R, JM --> R, dan JR --> K untuk KJRM; JM adalah satu-satunya kunci skema yang terakhir ini.

```
               -----------------
               |     KDJRMN    |
               |   kunci = JM  |
               | K-->D  KM-->N |
               | JR-->K JM-->R |
               |     DJ-->R    |
               -----------------
                /             \
     ------------          -----------------
     |   KMN    |          |     KDJRM     |
     |kunci = KM|          |  kunci =  MJ  |
     |  KM-->N  |          | K-->D  DJ-->R |
     ------------          | JR-->K JM-->R |
                           -----------------
                          /                 \
                 -----------      -------------------
                 |   KD    |      |      KJRM       |
                 |kunci = K|      |   kunci =  MJ   |
                 |  K-->D  |      | KJ-->R   JR-->K |
                 -----------      |     JM-->R      |
                                  -------------------
                                   /             \
                          ---------------    ------------
                          |    KJR      |    |   KJM    |
                          |kunci = KJ,JR|    |kunci = MJ|
                          |   KJ-->R    |    |  JM-->K  |
                          |   JR-->K    |    ------------
                          ---------------
```

Gambar 2-1. Pohon penguraian.

Perhatikan bahwa KJ --> R diperlukan dalam penutup daripada KJRM, meskipun di dalam KDJRM ia didapat dari K --> D dan DJ --> R. KD ada dalam bentuk normal Boyce-Codd, dan satu penguraian lagi daripada KJRM dengan mempergunakan KJ --> R, menempatkan skema databasis seluruhnya ke dalam bentuk yang diinginkan.
Masalah penguraian yang ditampilkan dalam gambar yalah bahwa gayutan fungsional DJ --> R tidak dipertahankan oleh penguraian di atas. Yalah, proyeksi F ke dalam KMN, KD, KJR, dan KJM, yang dapat disajikan oleh penutup

KM --> N	JR --> K	
K --> D	JM --> K	KJ ----> R

yang didapat dengan mengambil penutup minimal dalam setiap daun Gambar 2-1, tidak mengimplikasikan DJ --> R.

Penguraian ke dalam Bentuk Normal Ketiga (3NF) yang Mempertahankan Gayutan Fungsional.

Algoritme: Penguraian ke dalam Bentuk Normal Ketiga (3NF) yang mempertahankan Gayutan.

Masukan : Skema relasi R dan himpunan gayutan fungsional F, yang tanpa mengurangi ke-umumannya diandaikan penutup minimal.

Keluaran : Penguraian R yang mempertahankan gayutan fungsional sedemikian sehingga setiap skema relasi ada dalam bentuk normal ketiga terhadap proyeksi F ke dalam skema itu.

Metode :

Jika ada sebarang atribut daripada R yang tidak tersangkut dalam satu gayutan fungsional dalam F, baik di ruas kiri atau di ruas kanan, maka atribut itu, pada asanya, dapat membentuk suatu skema relasi sendirian, dan kita akan menghapuskannya dari R.
Jika satu daripada gayutan fungsional F menyangkut semua atribut R, maka keluarkan R itu sendiri. Bila tidak, maka keluaran penguraian p yang dikeluarkan terdiri dari skema XA untuk setiap gayutan fungsional X --> A dalam F. Tetapi, jika X --> A1, X --> A2, An daripada X Ai untuk 1 <= i <= n, dan memang penyulihan ini yang umumnya dipilih.

Contoh :

Ditinjau kembali skema relasi KDJRMN yang gayutan-gayutan fungsionalnya mempunyai penutup minimal :

K --> D	JD --> R	KM --> N
JR --> K	JM --> R	

Algoritma penguraian di atas menghasilkan himpunan skema relasi KD, KJR, JRD, KNM dan JRM.

Penguraian ke dalam Bentuk Normal Ketiga dengan Join Nirlesap dan Mempertahankan Gayutan-gayutan Fungsional.

Kita telah melihat bahwa kita dapat menguraikan setiap skema relasi R ke dalam himpunan skema p = {R1, ..., Rk} sedemikian sehingga p mempunyai join nirlesap dan setiap Ri ada dalam bentuk normal Boyce-Codd (dan dengan demikian dalam bentuk normal ketiga). Kita juga dapat menguraikan R ke dalam s = {S1,..., Sm} sedemikian sehingga s mempertahankan himpunan gayutan fungsional F, dan setiap Sj ada dalam bentuk normal ketiga.

Teorem II:

Misalkan s adalah penguraian dari R dalam bentuk normal ketiga yang disusun dengan Algoritme di atas, dan misalkan X adalah kunci untuk R. Maka r = s + {X} adalah penguraian R dengan semua skema relasi dalam bentuk normal ketiga : penguraian itu mempertahankan gayutan-gayutan fungsional dan mempunyai sifat join nirlesap.

Kita telah menguraikan secara sederhana dasar-dasar teori yang dipakai di dalam databasis relasional. Dalam bab berikut ini kita akan mencoba menerapkan teori tersebut untuk menyusun suatu databasis akademis.

3. Perancangan Databasis Akademis.

Bila merencanakan suatu databasis dengan mempergunakan model relasional, kita akan sering berhadapan dengan pilihan di antara alternatif himpunan skema relasi. Penting pada perencanaan skema-

skema databasis adalah pengertian gayutan (dependensi) data, yalah kendala pada relasi-relasi yang mungkin yang dapat merupakan nilai terakhir untuk suatu skema relasi.

3.1. Bahasa Pertanyaan Databasis Relasional: SQL.

Suatu bahasa yang sekurang-kurangnya dapat mensimulasi kalkulus tuple, atau aljabar relasional dikatakan lengkap. Dalam kenyataannya, bahasa-bahasa pengolah data umumnya memiliki kemampuan yang lebih daripada kalkulus relasional. Karena semua bahasa pengolah data mengandung perintah-perintah sisipan (insertion), penghapusan (deletion), dan perubahan (modification) yang bukan bagian daripada aljabar relasional ataupun kalkulus relasional.

Beberapa kemampuan tambahan yang sering dimiliki antara lain:

1. Kemampuan hitung (aritmatika):

   Atom dalam ungkapan-ungkapan kalkulus atau seleksi dalam ungkapan aljabar dapat mengandung hitungan atau komputasi aritmatik di samping perbandingan-perbandingan, seperti misalnya $A < (B+5)$.
2. Perintah-perintah penunjukan dan cetak (print).

   Bahasa-bahasa umumnya mengijinkan pencetakan relasi yang dibangun dengan ungkapan aljabar atau kalkulus, atau penunjukan dan pemberian harga suatu relasi dengan nilai relasi terhitung.
3. Fungsi kelompok atau gumpalan.

   Operasi-operasi seperti rerata, jumlah, minimum atau maksimum sering dapat diterapkan pada kolom suatu relasi untuk memperoleh suatu besaran tunggal.

Dengan demikian, bahasa-bahasa yang baik untuk dipergunakan dalam databasis relasional umumnya lebih daripada lengkap.

Bahasa SQL.

Bahasa SQL adalah bahasa pengolah data untuk komputer-komputer IBM tipe mainframe. Operasi dasar dalam SQL adalah pemetaan yang diungkapkan dengan bentuk SELECT-FROM-WHERE [DAT-81], [IBM-84].

```
SELECT list-atribut
       FROM list-tabel/relasi
          WHERE persyaratan.
```

List atribut adalah daftar atribut-atribut yang diinginkan nilainya. Atribut-atribut ini harus terdapat dalam tabel (relasi) yang disebutkan di dalam list-tabel pada baris kedua. Persyaratan yang muncul setelah kata WHERE dapat mengandung operator-operator perbandingan seperti: =, =/= (taksama), >, >=, <, atau <=; operator Boolean AND, OR, dan NOT, dan tanda kurung bila diperlukan untuk menyatakan urutan perhitungan.
Untuk membuat tabel (relasi) pertama kali dipakai perintah CREATE TABLE dengan bentuk umum sebagai berikut:

```
CREATE TABLE  namatabel (namakol-1  tipe-1 [not null]
              [, namakol-2  tipe-2 [not null]], ...)
              in  [nama-dbspace]
```

```
1) Nama Jurusan (JURUSAN)          10   Aksara
2) Nama Laboratorium (LABO)        10   Aksara
3) NIP Staf Pengajar (NIPDOS)      9+1  Aksara
4) Nama Staf Pengajar (NAMADOS)    40   Aksara
5) Nomer matakuliah (KODEMK)       6+1  Aksara
   huruf pertama- kode A penanggung jawab kuliah/praktikum
                       B pelaksana kuliah/praktikum
                       C asistensi kuliah/praktikum
6) Nama matakuliah (NAMAMK)        40   Aksara
7) Jumlah SKS (SKS)                4    Numerik
8) Semester (SEM)                  1    Numerik
9) Ruang Kuliah (RUANG)            4    Aksara
10) Kapasitas ruang (KAPAS)        3    Numerik
11) Waktu kuliah ( JAM)            2    Numerik
12) Hari kuliah (HARI)             6    Aksara
13) Ujian sisipan (SISIPAN)        6+2  Aksara
14) Ujian Akhir (UJIAN)            6+2  Aksara
15) Jabatan akademik (JABAKAD)     15   Aksara
16) Pangkat (GOLONGAN)             2    Aksara
17) Program Studi (PROSTUDI)       11   Aksara
```

Gambar 3-1. Nama-nama medan.

Teknik Perencaan Databasis.

Kita andaikan masukan ke dalam proses perencanaan databasis mengandung himpunan medan-medan data, dan pengertian tentang ikatan gayutan di antara mereka .

Akan dicoba merencanakan databasis pengajaran di fakultas MIPA : dengan nama-nama medan seperti tercantum dalam Gambar 3-1.

Gayutan-gayutan Fungsional (F.D) :

1) NIPDOS ---> NAMADOS, JABAKAD, GOLONGAN
2) NAMADOS ---> JURUSAN
3) KODEMK ---> NAMAMK, JURUSAN
4) KODEMK ---> SKS, SEM
5) KODEMK, NIPDOS ---> RUANG, JAM, HARI
6) RUANG ---> KAPAS
7) KODEMK, NIPDOS ---> SISIPAN, UJIAN
8) PROSTUDI --->> KODEMK
9) JURUSAN --->> LABO, PROSTUDI, NAMADOS
10) NAMADOS --->> KODEMK
11) LABO ---> JURUSAN

Gambar 3-2. Gayutan-gayutan Fungsional.

INF : NIPDOS, KODEMK * NAMADOS, NAMAMK, SKS, SEM, RUANG, KAPAS, HARI, JAM, SISIPAN, UJIAN, JABAKAD, GOLONGAN, PROSTUDI, JURUSAN, LABO.

Gambar 3-3. Bentuk Normal Pertama.

Andaikan NIP Dosen, Kode matakuliah, Nama Jurusan dan Nama Program Studi mempunyai nilai tunggal. Pada Gambar 3-2 ditulis gayutan fungsional yang menyatakan ikatan di antara medan-medan yang ditinjau. Pengawanan (ikatan) nomor (1), (2) adalah contoh ikatan ekan (gayutan uner), yang menunjukkan nilai satu medan di ruas kanan, yaitu untuk NIP Staf pengajar tertentu hanya ada satu nama yang bersangkutan dengannya. Pengawanan nomor (7) adalah variasi daripada pengawanan ekan, yang menunjukkan nilai daripada masing-masing dua medan diperlukan untuk mendapatkan nilai medan hasil yang tunggal. Pengawanan nomor (9) dan (10) adalah contoh pengawanan ganda (majemuk) (dengan panah kepala dua) yang di dalamnya nilai medan di ruas kiri dikawankan dengan nol, satu atau lebih nilai medan di ruas kanan, misalnya : nama laboratorium akan bersangkutan dengan beberapa nama staf pengajar dan seorang staf pengajar akan berkaitan dengan beberapa nama matakuliah yang diasuhnya dalam semester yang bersangkutan.

Teknik Penormalan Data.

Teknik perencanaan databasis yang pertama dinamakan **penormalan data**. Penormalan data adalah suatu prosedur untuk mengatur medan-medan ke dalam grup atau **tabel** atau **relasi** yang memiliki sifat-sifat tertentu.

Langkah-langkah berikut ini mengungkapkan intisari proses penormalan data:

Gambar 3-3 menunjukkan medan dalam Gambar 3-1 disusun dalam bentuk normal pertama (1NF). Mereka hanya didaftar dengan penunjukan himpunan minimum medan yang dapat menjadi kunci yang sah, ditunjukkan di sebelah kiri tanda bintang (*). Jadi, NIP Dosen dan Kode matakuliah, secara sendiri-sendiri atau bersama-sama, menentukan setiap medan yang lain dalam kumpulan itu. Satu medan sendirian ternyata tidak cukup menentukan. Hal ini dapat dilihat kebenarannya dari himpunan bagian pengawanan (ikatan) ekan dalam Gambar 3-2, ikatan nomor (1) - (7).

Bentuk normal pertama adalah titik awal saja. Jika data dimasukkan dalam format ini, terlihat akan banyak pengulangan data. Misalnya, karena data staf pengajar dan data matakuliah dicampur, maka nama staf pengajar, pangkat dan jabatan akademik akan terulang setiap kali NIP staf tersebut muncul.

Gambar 3-4 menunjukkan tiga relasi dalam bentuk normal kedua (2NF). Dalam setiap relasi, dilihat secara tersendiri, setiap medan bukan kunci gayut pada kunci keseluruhan, tidak hanya pada sebagian daripadanya. Dalam dua relasi yang pertama gayutan adalah trivial, karena kunci hanya terdiri dari satu medan, 2NF :

1) NIPDOS * NAMADOS, JABAKAD, GOLONGAN, LABO
2) KODEMK * NAMAMK, PROSTUDI, SKS, SEM, JURUSAN
3) KODEMK, NIPDOS, RUANG * JAM, HARI, KAPAS, SISPAN, UJIAN

Gambar 3-4. Bentuk Normal Kedua.

sedangkan dalam relasi ketiga kombinasi tiga medan diperlukan untuk memberi arti ujian sisipan atau ujian akhir semester. Pemecahan dari bentuk normal pertama menjadi bentuk normal kedua telah mengurangi pengulangan data.
Meskipun relasi di dalam Gambar 3-4 memenuhi kaidah-kaidah bentuk normal kedua, ia masih menghasilkan pengulangan data karena satu dari medan bukan kunci mendefinisikan salah satu medan bukan-kunci yang lain, misalnya : NAMADOS (nama staf pengajar) menentukan secara transitif LABO (nama laboratorium). Dengan menguraikan relasi pertama Gambar 3-4 menjadi dua relasi nomor (1) dan nomor (2) Gambar 3-5, ia telah mengurangi kelebihan data, dan membuatnya menjadi dalam bentuk normal ketiga (3NF).
Relasi nomor (2) Gambar 3-4 diuraikan menjadi relasi nomor (3) dan nomor (4) Gambar 3-5, dan relasi nomor (3) Gambar 3-4 diuraikan menjadi relasi nomor (5), (6) dan (7) Gambar 3-5. Mereka memenuhi kaidah bentuk normal ketiga, sehingga relasi-relasi dalam Gambar 3-5 adalah dalam bentuk normal ketiga.

3NF :

1) NIPDOS * NAMADOS, JABAKAD, GOLONGAN
2) NAMADOS * LABO
3) KODEMK * NAMAMK, SKS, SEM, JURUSAN
4) JURUSAN * PROSTUDI
5) KODEMK, NIPDOS * RUANG, HARI, JAM
6) RUANG * KAPAS
7) KODEMK, RUANG * UJIAN, SISIPAN

Gambar 3-5. Bentuk Normal Ketiga.

Sekarang marilah kita tinjau-ulang beberapa konsep databasis dalam kaitan perencanaan databasis relasional dalam Gambar 3-5. Pertama, kelebihan data di antara medan-medan bukan-kunci telah disingkirkan. Karena nomor kode matakuliah adalah tunggal, maka hanya ada satu pemunculan rekaman untuk suatu matakuliah dalam relasi nomor (3) Gambar 3-5. Dalam relasi itu muncul nama matakuliah, cacah SKS, ditawarkan pada semester ke berapa, dan termasuk dalam jurusan apa; dan ini merupakan satu-satunya pemunculan rekaman itu di dalam keseluruhan databasis.

Perhatian penting lain dalam databasis adalah penanganan hubungan majemuk di antara ujud-ujud yang dapat muncul : satu-ke-banyak dan banyak-ke-banyak. Tipe satu-ke-banyak disajikan dalam contoh oleh pengawanan nomor (9) dan nomor (10) Gambar 3-2, yang menunjukkan bahwa seorang staf pengajar (dosen) hanya bekerja di dalam satu Jurusan tetapi bahwa satu Jurusan mempunyai banyak staf pengajar. Nama staf pengajar, yang merupakan medan bukan-kunci dalam relasi nomor (1) Gambar 3-4, dinamakan kunci-asing, karena menjadi kunci dalam relasi nomor (2) dalam Gambar 3-5. Suatu **kunci-asing** adalah medan kunci atau medan bukan-kunci atau kombinasi medan-medan di dalam satu relasi yang merupakan kunci suatu relasi lain di dalam databasis itu.

Bila perhatian dipusatkan dalam relasi-relasi Gambar 3-5, menjadi jelas bahwa relasi (1) memungkinkan kita mengetahui keterangan tentang nama staf pengajar (dosen), jabatan akademis dan golongan pangkatnya, dan dari relasi (2) kita dapat mengetahui di laboratorium apa dia bekerja.

Untuk mengetahui nama-nama dosen yang mengajar setiap matakuliah, dapat dilakukan dengan menggabungkan relasi (5) dan (1).

Gayutan (9) dan (10) Gambar 3-2 menunjukkan adanya hubungan satu-ke-banyak dan banyak-ke-banyak. Hubungan satu-ke-banyak yang diungkapkan oleh gayutan fungsional (9) adalah bahwa satu Jurusan dapat memiliki banyak laboratorium, banyak program studi dan tentu saja banyak staf pengajar. Sedangkan gayutan fungsional (10) mengungkapkan hubungan banyak-ke-banyak karena seorang staf pengajar dapat memberikan (mengajar) beberapa matakuliah, dan sebaliknya suatu matakuliah dapat diberikan (diasuh) oleh beberapa staf pengajar (dosen).

3.3. Penyusunan Databasis Akademis.

Teknik Penormalan Data - Databasis Relasional.

Kita telah menunjukkan bahwa proses penormalan data menghasilkan himpunan relasi yang cocok dipergunakan sebagai data basis relasional.

Dalam keadaan yang sesungguhnya, perencanaan relasional - hasil daripada proses penormalan data - mungkin harus diubah untuk alasan-alasan unjuk-kerja dan hasil-kerja. Kenyataan ini membedakan apa yang biasanya dinamakan perencanaan databasis secara nalar dari fase berikutnya, perencanaan databasis fisis, yang dipergunakan untuk mengubah perencanaan databasis nalaran, dengan alasan unjuk-kerja tersebut.

Hubungan satu-ke-banyak dan Kunci Asing.

Dalam pendekatan penormalan data, hubungan satu-ke-banyak keluar secara alami sebagai hasil gayutan-gayutan fungsional medan dan langkah-langkah proses penormalan data. Adalah penting bahwa perencaa memehami konsep kunci asing sebagai jalan mempengaruhi hubungan satu-ke-banyak sebagai bagian daripada proses dalam pendekatan penormalan data.

Hubungan banyak-ke-banyak.

Dalam teknik penormalan data, hubungan banyak-ke-banyak ditangani secara alami bila kita mengenal bahwa data perpotongan harus mempunyai lebih dari hanya pengenal ujud (entitas) agar supaya dapat ditentukan secara penuh. Jika tidak ada data perpotongan untuk hubungan banyak-ke-banyak tersebut, maka sebagai akibatnya langkah tambahan akan diperlukan untuk mengenal hubungan itu dari daftar gayutan fungsional.

Dalam kerangka struktur databasis, struktur databasis relasional mensyaratkan bahwa hubungan banyak-ke-banyak disajikan dengan cara tabel.

Dari hasil analisis pada pasal 3.2 di atas menjadi jelaslah bahwa untuk perencanaan dengan nama-nama medan : NIP dosen (NIPDOS), Nama dosen (NAMADOS), Jabatan Akademik (JABAKAD), Golongan (GOLONGAN), Kode Matakuliah (KODEMK), Nama Matakuliah (NAMAMK), SKS (SKS), Semester (SEM), Program Studi (PROSTUDI), Nama Jurusan (JURUSAN), Nama Laboratorium (LABO), Kode Ruang (RUANG), Kapasitas Ruang (KAPAS), Jadwal Hari (HARI), Jam Kuliah (JAM), Tanggal Ujian Sisipan (SISIPAN), dan Tanggal Ujian Akhir (UJIAN), telah tersusun relasi (tabel) yang memenuhi bentuk normal ketiga (3NF) sebagai berikut:

DAFDOS (NIPDOS, NAMADOS, JABAKAD, GOLONGAN)
DOSLAB (NAMADOS, LABO)
DAFMAKU (KODEMK, NAMAMK, SKS, SEM, JURUSAN)
DAFPROS (JURUSAN, PROSTUDI)
JADWAL (KODEMK, NIPDOS, RUANG, HARI, JAM)
DAFRUANG (RUANG, KAPASITAS)
JADWTES (KODEMK, RUANG, SISIPAN, UJIAN)

Relasi-relasi ini kemudian dapat dipakai sebagai isi daripada databasis akademis yang implementasinya dapat dikerjakan dengan bahasa SQL.

4. PEMBAHASAN DAN PENUTUP.

Hasil penguraian databasis akademis yang ditinjau dalam pasal 3.3 di atas, menghasilkan tabel-tabel (relasi-relasi): DAFDOS, DOSLAB, DAFMAKU, DAFPROS, JADWAL, DAFRUANG, dan JADWTES.

Dari tabel-tabel yang telah tercipta tersebut, kita dapat membuat kluaran yang memenuhi permintaan kita. Misalnya, untuk membuat daftar nama dosen dengan nama matakuliah yang diasuhnya, kita dapat memproyeksikan dengan perintah SELECT dari gabungan tabel DAFDOS, DAFMAKU dan JADWAL. Untuk membuat tabel jadwal ujian yang berisi kode matakuliah, nama matakuliah, nama dosen, tanggal ujian, ruang dan jam ujian, kita dapat mengambilnya dari gabungan tabel JADWTES, JADWAL, DAFMAKU dan DAFDOS. Bila tabel hasil ini sangat diperlukan, maka dapat dibuatkan tabel tersendiri sebagai pandangan (VIEW) terhadap tabel-tabel tersebut di atas, yang secara otomatis akan mencerminkan setiap perubahan yang terjadi di dalam tabel-tabel aslinya.

Kami mengucapkan terimakasih kepada Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam yang telah membantu terlaksananya penelitian ini dengan dana DPP/SPP FMIPA UGM. Mudah-mudahan usaha ini dapat berkelanjutan dalam mendorong penelitian baik yang bersifat murni maupun bersifat pengembangan dan terapan. Juga kepada UPT KOMPUTER UGM yang telah menyediakan fasilitas pengolahan dengan komputer sehingga penelitian ini dapat terwujudkan.

5. DAFTAR PUSTAKA/ACUAN.

[ALA-86] Alagic, Suad, 1986, "Relational Database Technology", Springer-Verlag, Berlin.

[ARM-74] Armstrong, W.W., 1974, "Dependency structures of database relationships", Proc. 1974 IFIP Congress, hal.580-583, North-Holland, Amsterdam.

[BEE-86] Beeri, C., M. Kifer, 1986, "An Integrated Approach to Logical Design of Relational Database Schemes", ACM TODS vol.11 no.2, hal.134-158, Juni.

[CER-86] Ceri, S., G. Gottlob, 1986, "Normalization of relation and PROLOG", CACM vol.29, 6 (Juni 1986), hal.524-544.

[COD-70] Codd, E.F., 1970, "A Relational Model of Data for Large shared Data Banks", Commun. ACM vol.13 no.6, hal.377-387, Juni.

[CODD-72] Codd, E.F.,1972, "Relational Completeness of Database Sublanguages", dalam Data Base Systems, Courant Computer Science Symposia Series, vol.6, Prentice-Hall, Englewood Cliffs, N.J.

[DAH-82] Dahl, V., 1982,"On Database Systems Development through Logic", ACM TODS, vol.7 no.1, hal.102-123, Maret.

[DAT-81] Date, C.J., 1981,"An Introduction to Database Systems", edisi ke-3, Addison Wesley, Singapore.

[DEL-85] Delobel, C., M. Adiba, 1985, "Relational Database Systems", Elsevier Science Publ. Co., New York.

[DEL-79] Delobel, C., 1979, "Theoretical Aspects of Modeling in Relational Database", Rapport de Recherche, R.R.No.177, IMAG, USM Grenoble, Perancis.

[DEN-79] Denning, D.E., P.J. Denning, 1979, "Data Security", ACM

Computing Surveys, vol.11 no.3, hal.227-249, September.

[FED-87] Federowics, J., 1987, "Database Performance Evaluation in an Indexed File Environment", ACM TODS, vol.12 no.1, hal.85-110, Maret.

[GIL-87] Gillenson, M.L., 1987, "The Duality of Database Structures and Design Techniques", Commun. ACM vol.30 no.12, hal.1056-1065, Desember.

[HOR-77] Horowitz, E., S. Sahni, 1977, "Fundamentals of Data Structures", Pitman, London.

[JAR-85] Jarke, M., Y. Vassiliou, 1985, "A Framework for Choosing a Database Query language", ACM Comp. Surveys vol. 17 no.3, hal.313-339, September.

[LIE-81] Lien, Y.E., 1981, "Hierarchical Schemata for Relational Database", ACM TODS, vol.6 no.1, hal.48-69, Maret.

[MAR-83] March, S.T., 1983, "Techniques for Structuring Database Records", ACM Comp. Surveys vol.15 no.1, hal.45-79, Maret.

[OSB-83] Osborn, S.L., T.E. Heaven, 1983, "The Design of a Relational Database System with Abstract Data Types for Domains", ACM TODS vol.11 no.3, hal.357-373, September.

[PRA-88] Prawirosusanto, Sumartono, 1988, "Pengembangan Penerapan Databasis Relasional", Laporan Penelitian, FMIPA UGM, Departemen Peendidikan dan Kebudayaan, Yogyakarta.

[TRE-76] Trembley, J.P., P.G. Sorenson, 1976, "An Introduction to Data Structures with Applications", McGraw-Hill Kogakusha, Tokyo.

[ULL-82] Ullman, J.D., 1982, "Principles of Database Systems", edisi ke-2, Computer Science Press, Rockville, Maryland.

[VER-78] Verhofstad, J.S.M., 1978, "Recovery Techniques for Database Systems", ACM Comp. Surveys vol.10 no.2, hal.167-195, Juni.

[WON-79] Won Kim, 1979, "Relational Database Systems", ACM Comp. Surveys vol.11 no.3, hal.185-211, September.

[WON-82] Won Kim, 1982, "On Optimizing an SQL-like Nested Query" ACM TODS vol.7 no.3, hal.443-469, September.

[ZAN-81] Zaniolo, C., M.A. Melkanoff, 1981, "On the Design of Relational Database Schemata", ACM Trans. on Database

Systems, vol.6 no.1, hal.1-47, Maret.

[ZAN-82] Zaniolo, C., 1982, "A New Normal Form for the Design of Relational Database Schemata", ACM TODS vol.7 no.3, hal.489-499, September.